UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/517,462	06/08/2005	Hartmut Pettendorf	P16483-US1	1351

27045 7590 02/18/2009
ERICSSON INC.
6300 LEGACY DRIVE
M/S EVR 1-C-11
PLANO, TX 75024

EXAMINER	
BOKHARI, SYED M	
ART UNIT	PAPER NUMBER
2416	
MAIL DATE	DELIVERY MODE
02/18/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/517,462	**Applicant(s)** PETTENDORF ET AL.	
	Examiner SYED BOKHARI	**Art Unit** 2416	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11/10/2009*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,8,10,12,13,17,19 and 21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,8,10,12,13,17,19 and 21* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.☒ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

Response to Amendment

1. Applicant's amendment filed on November 10th, 2008 has been entered. Claims 1, 8, 10, 12, 17, 19 and 21 have been amended. Claims 3-7, 9, 11, 14-16, 18 and 20 have been canceled. Claims 1-2, 8, 10, 12-13, 17, 19 and 21 are pending in the application.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

4. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein

were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

5. Claims 1, 10 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Misra et al. (US 2002/0009127 A1) in view of Fulvio (WO 01/50658 A1), Piccinonno (WO 01/50659 A1), Jechoux et al. (US 2002/0041636 A1), Kim et a. (US 6,671,251 B1) and further in view of Eo et al. (US 6,069,574).

Misra et al. disclose a efficient spreader for spread spectrum communication system with the following features: regarding claim 1, a code generator for generating an orthogonal code having a spreading factor (SF) and an index (k) (Fig. 5, a system structure, see "the spreader 17 comprises a code generator 21" recited in paragraph 0052 lines 1-7 and paragraph 0053 lines 1-7), wherein the spreading factor (SF) is selectable from values in a range 1<SF< or =.SFmax with Sfmax denoting a fixed maximum spreading factor and wherein the index (k) is in a range 0<k< or =Sfmax-1, the code generator comprising an index conversion unit for converting the index (k) into a modified index (j) associated with a corresponding code having the fixed maximum spreading factor wherein the modified index (i) is in a range 0<i<or =Sfmax-1 (Fig. 6a-d, control flow diagrams, see "form a periodic long code to the maximum spreading factor (Sfmax) of the communication system" recited in paragraph 0064 lines 1-13 and

paragraph 0072 line 1); regarding claim 10, parallel code generator for concurrently generating a number p>1 orthogonal codes having respective spreading factors (SF.sub.1, .SF.sub.p) and indices (k.sub.1, .k.sub.p) (Fig. 5, a system structure, see “the spreader 17 comprises a code generator 21” recited in paragraph 0052 lines 1-7 and paragraph 0053 lines 1-7), wherein he spreading factors are selectable from values in a range 1<SF.1, .F.p.< or =.SF.max, with SF.max denoting a fix maximum spreading factor the parallel code generator comprising, a number (p) of code generators, each of the code generators including an index conversion unit for converting the index (k) into a modified index (j) associated with a corresponding code having the fixed maximum spreading factor wherein the modified index (i) is in a range 0<i<or =Sfmax-1 (Fig. 6a-d, control flow diagrams, see “form a periodic long code to the maximum spreading factor (Sfmax) of the communication system” recited in paragraph 0064 lines 1-13 and paragraph 0072 line 1); regarding claim 12, a method of generating an orthogonal code having a spreading factor (SF) and an index (k) (Fig. 5, a system structure, see “the spreader 17 comprises a code generator 21” recited in paragraph 0052 lines 1-7 and paragraph 0053 lines 1-7) and wherein the spreading factor (SF) is selectable from values in a range 1<SF< or =.Sfmax, with Sfmax denoting a fixed maximum spreading factor wherein the index (k) is in a range 0<k< or =Sfmax-1, the method comprising the steps of converting by an index conversion unit the index (k) into a modified index (j) associated with a corresponding code having the fixed maximum spreading wherein the modified index (i) is in a range 0<i<or =Sfmax-1 (Fig. 6a-d, control flow diagrams, see “form a periodic long code to the maximum spreading factor (Sfmax) of the

communication system" recited in paragraph 0064 lines 1-13 and paragraph 0072 line 1).

Misra et al. do not disclose the following features: regarding claim 1 a logic unit for solely performing logic operations on bits of the modified index (j) and bits of a counter value (i), thereby generating a code bit of the orthogonal code, counter for incrementing the counter value (i) and control means for causing the index conversion unit and the logic unit to sequentially repeat their operations utilizing counter values (i) incremented steps of one (1) until a desired number of code bits has been generated, a shift register for receiving and storing the index (k) in binary representation and receiving the number (s) and to shift the stored index (k) by (s) bit positions in the direction of more significant bit positions, multiplication means for multiplying the index (k) with a value of SF.sub.max/SF and the multiplication means comprising a mapping unit for mapping the spreading factor (SF) to a number (s) equal to log.sub.2{SF.max/SF}, a permutation unit for permuting the bits of the index (k), selection means for selecting, depending upon a mode signal indicating a desired type of the orthogonal code and the output of the permutation unit or the output of the shift register, thereby generating the modified index (j), wherein the logic unit includes adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits and combining means for combining the binary output values into the code bi; regarding claim 10, a logic unit for solely performing logic operations on bits of the modified index (j) and bits of a counter value (i), thereby generating a code bit of the orthogonal code, counter for incrementing the counter value (i) and control means for causing the index conversion unit and the logic unit to sequentially repeat their operations utilizing counter values (i) incremented steps of one (1) until a desired number of code bits has been generated, a shift register for receiving and storing the index (k) in binary representation and receiving the number (s) and to shift the stored index (k) by (s) bit positions in the direction of more significant bit positions, multiplication means for multiplying the index (k) with a value of SF.sub.max/SF and the multiplication means comprising a mapping unit for mapping the spreading factor (SF) to a number (s) equal to log.sub.2{SF.max/SF}, a permutation unit for permuting the bits of the index (k), selection means for selecting, depending upon a mode signal indicating a desired type of the orthogonal code and the output of the permutation unit or the output of the shift register, thereby generating the modified index (j), wherein the logic unit includes adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits and combining means for combining the binary output values into the code bi. Regarding claim 12, initializing a counter, value (i), solely performing logic operations on bits of the modified index (j) and bits of the counter value (i), thereby generating a code bit of the orthogonal code; d);

Fulvio discloses a an innovative generator for orthogonal variable spreading factor (OSVSF) with the following features: regarding claim 1, a logic unit for solely performing logic operations on bits of the modified index (j) and bits of a counter value (i), thereby generating a code bit of the orthogonal code (Fig. 1, generator for orthogonal spreading sequence, see “the logical unit AND has two input valves of index k and counter value” recited in lines 19-24 see “the desired page 1), counter for incrementing the counter value (i) (Fig. 1, generator for orthogonal spreading sequence, sequence is generated when counter runs” recited in lines 24-25 page 1), control means for causing the index conversion unit and the logic unit to sequentially repeat their operations utilizing counter values (i) incremented steps of one (1) until a desired number of code bits has been generated (Fig. 1, generator for orthogonal spreading sequence, see “the logical unit AND has two input valves of index k and counter value” recited in lines 19-24 page 1); regarding claim 10, a logic unit for solely performing logic operations on bits of the modified index (j) and bits of a counter value (i), thereby generating a code bit of the orthogonal code (Fig. 1, generator for orthogonal spreading sequence, see “the logical unit AND has two input valves of index k and counter value” recited in lines 19-24 page 1), counter for incrementing the counter value (i) (Fig. 1, generator for orthogonal spreading sequence, see “the desired sequence is generated when counter runs” recited in lines 24-25 page 1), control means for causing the index conversion unit and the logic unit to sequentially repeat their operations utilizing counter values (i) incremented steps of one (1) until a desired number of code bits has been generated (Fig. 1, generator for orthogonal spreading sequence, see “the logical unit AND has two input valves of index k and counter value” recited in lines 19-24 page 1); regarding claim 12, initializing a counter, value (i), solely performing logic operations on bits of the modified index (j) and bits of the counter value (i), thereby generating a code bit of the orthogonal code (Fig. 1, generator for orthogonal spreading sequence, see “the logical unit AND has two input valves of index k and counter value” recited in lines 19-24 page 1) incrementing the counter value (i) by one (Fig. 1, generator for orthogonal spreading sequence, see “the desired sequence is generated when counter runs” recited in lines 24-25 page 1) and e) repeating steps c) and d) until a desired number of code bits has been generated (Fig. 1, generator for orthogonal spreading sequence, see “the logical unit AND has two input valves of index k and counter value” recited in lines 19-24 page 1), repeating steps until a desired number of code bits has been generated (Fig. 1, generator for orthogonal spreading sequence, see “the desired sequence is generated when counter runs” recited in lines 24-25 page 1), control means for causing the index conversion unit and the logic unit to sequentially repeat their operations utilizing counter values (i) incremented steps of one (1) until a desired number of code bits has been generated (Fig. 1, generator for orthogonal spreading sequence, see “the logical unit AND has two input valves of index k and counter value” recited in lines 19-24 page 1);

It would have been obvious to one of the ordinary skill in the art at the time of invention to modify the system of Misra et al. by using the features, as taught by Fulvio, in order to provide a logic unit for performing logic operations on bits of the modified index (j) and bits of a counter value (i), thereby generating a code bit of the orthogonal code and a counter for generating the counter value (i); wherein each of the code generators generates one of the (p) orthogonal codes having a particular one of the spreading factors and a particular one of the indices. The motivation of using these functions is to enhance the system in a cost effective manner.

Misra et al. and Fulvio do not disclose the following features: regarding claim 1, a shift register for receiving and storing the index (k) in binary representation and receiving the number (s) and to shift the stored index (k) by (s) bit positions in the direction of more significant bit positions, multiplication means for multiplying the index (k) with a value of SF.sub.max/SF and the multiplication means comprising a mapping unit for mapping the spreading factor (SF) to a number (s) equal to log.sub.2{SF.max/SF}, a permutation unit for permuting the bits of the index (k), selection means for selecting, depending upon a mode signal indicating a desired type of the orthogonal code and the output of the permutation unit or the output of the shift register, thereby generating the modified index (j), wherein the logic unit includes adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits and combining means for combining the binary output values into the code bi; regarding claim 10, a shift register for receiving and storing the index (k) in binary representation and receiving the number (s) and to shift the stored index (k) by (s) bit positions in the direction of more significant bit positions, multiplication means for multiplying the index (k) with a value of SF.sub.max/SF and the multiplication means comprising a mapping unit for mapping the spreading factor (SF) to a number (s) equal to log.sub.2{SF.max/SF}, a permutation unit for permuting the bits of the index (k), selection means for selecting, depending upon a mode signal indicating a desired type of the orthogonal code and the output of the permutation unit or the output of the shift register, thereby generating the modified index (j), wherein the logic unit includes adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits and combining means for combining the binary output values into the code bi; regarding claim 12, performing a shift register operation comprising receiving and storing the index (k) in binary representation and receiving the number (s) and to shift the stored index (k) by (s) bit positions in the direction of more significant bit positions, the converting step including multiplication means for multiplying the index (k) with a value of SF.sub.max/SF and the multiplication means comprising a mapping unit for mapping the spreading factor (SF) to a number (s) equal to log.sub.2{SF.max/SF}, a permutation unit for permuting the bits of the index (k), selection means for selecting, depending upon a mode signal indicating a desired type of the orthogonal code and the output of the permutation unit or the output of the shift register, thereby generating the modified index (j), wherein the logic unit includes adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits and combining means for combining the binary output values into the code bit.

Jechoux et al. disclose a communication system for channel estimation sequence with the following features: regarding claim 1, a shift register for receiving and storing the index (k) in binary representation (Fig. 3, bock diagram discrete in time of the transmission chain between the transmitter and the receiver, see ”shift register having I boxes in series referenced by index k” recited in paragraph 0003 lines 6-9), receiving the number (s) and to shift the stored index (k) by (s) bit positions in the direction of more significant bit positions (Fig. 3, bock diagram discrete in time of the transmission chain between the transmitter and the receiver, see ”contents slide towards right each time a symbol arrives at the entry” recited in paragraph 0003 lines 9-11); regarding claim 10, a shift register for receiving and storing the index (k) in binary representation (Fig. 3, bock diagram discrete in time of the transmission chain between the transmitter and the receiver, see ”shift register having I boxes in series referenced by index k” recited in paragraph 0003 lines 6-9) and receiving the number (s) and to shift the stored index (k) by (s) bit positions in the direction of more significant bit positions (Fig. 3, bock diagram discrete in time of the transmission chain between the transmitter and the receiver, see ”contents slide towards right each time a symbol arrives at the entry” recited in paragraph 0003 lines 9-11); regarding claim 12, performing a shift register operation comprising receiving and storing the index (k) in binary representation (Fig. 3, bock diagram discrete in time of the transmission chain between the transmitter and the receiver, see ”shift register having I boxes in series referenced by index k” recited in paragraph 0003 lines 6-9) and receiving the number (s) and to shift the stored index (k) by (s) bit positions in the direction of more significant bit positions (Fig. 3, bock diagram discrete in time of the transmission chain between the transmitter and the receiver, see ”contents slide towards right each time a symbol arrives at the entry” recited in paragraph 0003 lines 9-11);

It would have been obvious to one of the ordinary skill in the art at the time of invention to modify the system of Misra et al. and Fulvio by using the features, as taught by Jechoux et al., in order to provide a shift register adapted to receive and store the index (k) in binary representation, receiving the number (s) and to shift the stored index (k) by (s) bit positions in the direction of more significant bit positions. The motivation of using these functions is to enhance the system in a cost effective manner.

Misra et al., Fulvio and Jechoux et al. do not disclose the following features: regarding claim 1, the index converting step including multiplication means for multiplying the index (k) with a value of SF.sub.max/SF and the multiplication means comprising a mapping unit for mapping the spreading factor (SF) to a number (s) equal to log.sub.2{SF.max/SF}, a permutation unit for permuting the bits of the index (k), selection means for selecting, depending upon a mode signal indicating a desired type of the orthogonal code and the output of the permutation unit or the output of the shift register, thereby generating the modified index (j), wherein the logic unit includes adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits and combining means for combining the binary output values into the code bi; regarding claim 10, the index converting step including multiplication means for multiplying the index (k) with a value of SF.sub.max/SF and the multiplication means comprising a mapping unit for mapping the spreading factor (SF) to a number (s) equal to log.sub.2{SF.max/SF}, a permutation unit for permuting the bits of the index (k), selection means for selecting, depending upon a mode signal indicating a desired type of the orthogonal code and the output of the permutation unit or the output of the shift register, thereby generating the modified index (j), wherein the logic unit includes adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits and combining means for combining the binary output values into the code bi; regarding claim 12, the converting step including multiplication means for multiplying the index (k) with a value of SF.sub.max/SF and the multiplication means comprising a mapping unit for mapping the spreading factor (SF) to a number (s) equal to log.sub.2{SF.max/SF}, a permutation unit for permuting the bits of the index (k), selection means for selecting, depending upon a mode signal indicating a desired type of the orthogonal code and the output of the permutation unit or the output of the shift register, thereby generating the modified index (j), wherein the logic unit includes adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits and combining means for combining the binary output values into the code bit.

Piccinonno discloses a communication system for orthogonal variable spreading factor and Hadamard matrices generation with the following features: regarding claim 1, multiplication means for multiplying the index (k) with a value of SF.sub.max/SF, (Fig. 2, flow chart describing the a spreading sequence generator, see "a generator for rows of Hadamard matrices of dimension SF by an index n of k" recited in lines 21-27 on page 1) and the multiplication means comprising a mapping unit for mapping the spreading factor (SF) to a number (s) equal to log.sub.2{SF.max/SF} (Fig. 2, flow chart describing the a spreading sequence generator, see "if necessary n is transformed as described obtaining the binary number" recited in lines 10-14 on page 3); regarding claim 10, multiplication means for multiplying the index (k) with a value of SF.sub.max/SF, (Fig. 2, flow chart describing the a spreading sequence generator, see "a generator for rows of Hadamard matrices of dimension SF by an index n of k" recited in lines 21-27 on page 1) and the multiplication means comprising a mapping unit for mapping the spreading factor (SF) to a number (s) equal to log.sub.2{SF.max/SF} (Fig. 2, flow chart describing the a spreading sequence generator, see "if necessary n is transformed as described obtaining the binary number" recited in lines 10-14 on page 3); regarding claim 12, the converting step including multiplication means for multiplying the index (k) with a value of SF.sub.max/SF (Fig. 2, flow chart describing the a spreading sequence generator, see "a generator for rows of Hadamard matrices of dimension SF by an index n of k" recited in lines 21-27 on page 1) and the multiplication means comprising a mapping unit for mapping the spreading factor (SF) to a number (s) equal to log.sub.2{SF.max/SF} (Fig. 2, flow chart describing the a spreading sequence generator, see "if necessary n is transformed as described obtaining the binary number" recited in lines 10-14 on page 3).

It would have been obvious to one of the ordinary skill in the art at the time of invention to modify the system of Misra et al. with Fulvio and Jechoux et al. by using the features, as taught by Piccinonno, in order to provide multiplication means for multiplying the index (k) with a value of SF.sub.max/SF, the multiplication means comprising a mapping unit for mapping the spreading factor (SF) to a number (s) equal to log.sub.2{SF.max/SF}, Kim et al. discloses a communication system for generating complex quasi-orthogonal code for spreading channel data with the following features. The motivation of using these functions is to enhance the system in a cost effective manner.

Misra et al., Fulvio, Jechoux et al. and Piccinonno do not disclose the following features: regarding claim 1, a permutation unit for permuting the bits of the index (k), selection means for selecting, depending upon a mode signal indicating a desired type of the orthogonal code and the output of the permutation unit or the output of the shift register, thereby generating the modified index (j), wherein the logic unit includes adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits and combining means for combining the binary output values into the code bit regarding claim 10, a permutation unit for permuting the bits of the index (k), selection means for selecting, depending upon a mode signal indicating a desired type of the orthogonal code and the output of the permutation unit or the output of the shift register, thereby generating the modified index (j), wherein the logic unit includes adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits and combining means for combining the binary output values into the code bit; regarding claim 12, a permutation unit for permuting the bits of the index (k), selection means for selecting, depending upon a mode signal indicating a desired type of the orthogonal code and the output of the permutation unit or the output of the shift register, thereby generating the modified index (j), wherein the logic unit includes adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits and combining means for combining the binary output values into the code bit.

Kim et al. discloses a communication system for generating complex quasi-orthogonal code for spreading channel data with the following features: regarding claim 1, a permutation unit for permuting the bits of the index (k) (Fig. 5, a procedure for generating complex quasi-orthogonal codes, see “the M-sequence are column permuted with permutation function to generate a Walsh code” recited in column 5 lines 30-35 and column 8 lines 36-45), selection means for selecting, depending upon a mode signal indicating a desired type of the orthogonal code (Fig. 10, complex quasi-orthogonal code generator which generates quasi-orthogonal code mask in number, see “selectively outputs a mask corresponding to quasi-orthogonal code” recited in column 22 lines 12-20), the output of the permutation unit or the output of the shift register, thereby generating the modified index (j) (Fig. 5, a procedure for generating complex quasi-orthogonal codes, see “by circularly shifting M-sequence and column permutating the circularly shifted specific sequences for converting the generated M-sequence to Walsh orthogonal codes” recited in column 2 lines 53-65); regarding claim 10, a permutation unit for permuting the bits of the index (k) (Fig. 5, a procedure for generating complex quasi-orthogonal codes, see “the M-sequence are column permuted with permutation function to generate a Walsh code” recited in column 5 lines 30-35 and column 8 lines 36-45), selection means for selecting, depending upon a mode signal indicating a desired type of the orthogonal code (Fig. 10, complex quasi-orthogonal code generator which generates quasi-orthogonal code mask in number, see “selectively outputs a mask corresponding to quasi-orthogonal code” recited in column 22 lines 12-20), the output of the permutation unit or the output of the shift register, thereby generating the modified index (j) (Fig. 5, a procedure for generating complex quasi-orthogonal codes, see “by circularly shifting M-sequence and column permutating the circularly shifted specific sequences for converting the generated M-sequence to Walsh orthogonal codes” recited in column 2 lines 53-65); regarding claim 12, a permutation unit for permuting the bits of the index (k) (Fig. 5, a procedure for generating complex quasi-orthogonal codes, see “the M-sequence are column permuted with permutation function to generate a Walsh code” recited in column 5 lines 30-35 and column 8 lines 36-45), selection means for selecting, depending upon a mode signal indicating a desired type of the orthogonal code (Fig. 10, complex quasi-orthogonal code generator which generates quasi-orthogonal code mask in number, see "selectively outputs a mask corresponding to quasi-orthogonal code" recited in column 22 lines 12-20), and the output of the permutation unit or the output of the shift register, thereby generating the modified index (j) (Fig. 5, a procedure for generating complex quasi-orthogonal codes, see "by circularly shifting M-sequence and column permutating the circularly shifted specific sequences for converting the generated M-sequence to Walsh orthogonal codes" recited in column 2 lines 53-65);

It would have been obvious to one of the ordinary skill in the art at the time of invention to modify the system of Misra et al. with Fulvio, Jechoux et al. and Piccinonno by using the features, as taught by Kim et al, in order to provide a permutation unit for permuting the bits of the index (k), selection means for selecting, depending upon a mode signal indicating a desired type of the orthogonal code, the output of the permutation unit or the output of the shift register, thereby generating the modified index (j). The motivation of using these functions is to enhance the system in a cost effective manner.

Misra et al., Fulvio, Jechoux et al., Piccinonno and Kim et al. do not disclose the following features: regarding claim 1, wherein the logic unit includes adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits and combining means for combining the binary output values into the code bit; regarding claim 10, wherein the logic unit includes adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits and combining means for combining the binary output values into the code bit; regarding claim 12, wherein the logic unit includes adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits and combining means for combining the binary output values into the code bit.

Eo et al. discloses a Hadamard code generation circuit with the following features: regarding claim 1, wherein the logic unit includes adding means for performing binary AND operations (Fig. 1, a circuit for generating Hadamard codes, see "final Hadamard code is generated through a binary addition operation at AND gate" recited in column 1 lines 58-63), wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits (Fig. 1, a circuit for generating Hadamard codes, see "two AND gates 118 and 119 receive output signal from the counter 112 and index bit from the register 114" recited in column 1 lines 41-44); and combining means for combining the binary output values into the code bit (Fig. 1, a circuit for generating Hadamard codes, see "a XOR gate 124hereby generating a Hadamard code" recited in column 1 lines 44-46); regarding claim 10, wherein the logic unit includes adding means for performing binary AND operations (Fig. 1, a circuit for generating Hadamard codes, see "final Hadamard code is generated through a binary addition operation at AND gate" recited in column 1 lines 58-63), wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits (Fig. 1, a circuit for generating Hadamard codes, see "two AND gates 118 and 119 receive output signal from the counter 112 and index bit from the register 114" recited in column 1 lines 41-44) and combining means for combining the binary output values into the code bit (Fig. 1, a circuit for generating Hadamard codes, see "a XOR gate 124hereby generating a Hadamard code" recited in column 1 lines 44-46); regarding claim 12, wherein the logic unit includes adding means for performing binary AND operations (Fig. 1, a circuit for generating Hadamard codes, see "final Hadamard code is generated through a binary addition operation at AND gate" recited in column 1 lines 58-63), wherein the logic unit receive a bit of the modified index (j) and a bit of the counter value (i), and outputs a binary output value representing a binary AND combination of the two bits (Fig. 1, a circuit for generating Hadamard codes, see "two AND gates 118 and 119 receive output signal from the counter 112 and index bit from the register 114" recited in column 1 lines 41-44) and combining the binary output values into the code bit (Fig. 1, a circuit for generating Hadamard codes, see "a XOR gate 124hereby generating a Hadamard code" recited in column 1 lines 44-46).

It would have been obvious to one of the ordinary skill in the art at the time of invention to modify the system of Misra et al. with Fulvio, Jechoux et al., Piccinonno and Kim et al. by using the features, as taught by Eo et al, in order to provide wherein the logic unit includes adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits and combining means for combining the binary output values into the code bit. The motivation of using these functions is to enhance the system in a cost effective manner.

6. Claims 2 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Misra et al. (US 2002/0009127 A1) in view of Fulvio (WO 01/50658 A1) as applied to claims 1 and 12 above, and further in view of Kim et al. (USP 6,512,753 B1).

Misra et al. and Fulvio disclose the claimed limitations as described in paragraph 5 above. Misra et al. and Fulvio do not disclose the following features: regarding claim 2, wherein the corresponding code is one of: an orthogonal variable spreading factor (OVSF) code, a Hadamard code, and a Walsh code and regarding claim 13, wherein the corresponding code is one of: an orthogonal variable spreading factor (OVSF) code, a Hadamard code, and a Walsh code.

Kim et al. discloses an Orthogonal Variable Spreading Factor (OVSF) generator with the following features: regarding claim 2, wherein the corresponding code is one of:

an orthogonal variable spreading factor (OVSF) code, a Hadamard code, and a Walsh code (Fig. 7, a spreading device for a transmitter in a mobile communication system, see “adder 748 adds quasi-orthogonal mask and the Walsh orthogonal code from Walsh orthogonal code 746, to generate a quasi-orthogonal code” recited in column 6 lines 25-37); and regarding claim 13, wherein the corresponding code is one of: an orthogonal variable spreading factor (OVSF) code, a Hadamard code, and a Walsh code (Fig. 7, a spreading device for a transmitter in a mobile communication system, see “adder 748 adds quasi-orthogonal mask and the Walsh orthogonal code from Walsh orthogonal code 746, to generate a quasi-orthogonal code” recited in column 6 lines 25-37).

It would have been obvious to one of the ordinary skill in the art at the time of invention to modify the system of Misra et al. with Fulvio by using the features, as taught by Kim et al., in order to provide the corresponding code is one of: an orthogonal variable spreading factor (OVSF) code, a Hadamard code, and a Walsh code. The motivation of using this function is to enhance the system in a cost effective manner.

7. Claims 8 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Misra et al. (US 2002/0009127 A1) in view of Fulvio (WO 01/50658 A1) as applied to claims 1 and 12 above, and further in view of Eo et al. (US 6,069,574).

Misra et al. and Fulvio disclose the claimed limitations as described in paragraph 5 above. Misra et al. and Fulvio do not disclose the following features: regarding claim 8, wherein the combining means includes means for performing binary XOR operations and regarding claim 19, wherein the step of combining includes performing binary XOR operations.

Eo et al. discloses a Hadamard code generation circuit with the following features: regarding claim 8, wherein the combining means includes means for performing binary XOR operations (Fig. 1, a circuit for generating Hadamard codes, see "a XOR gate 124 hereby generating a Hadamard code" recited in column 1 lines 44-46); regarding claim 19, wherein the step of combining includes performing binary XOR operations (Fig. 1, a circuit for generating Hadamard codes, see "a XOR gate 124hereby generating a Hadamard code" recited in column 1 lines 44-46).

It would have been obvious to one of the ordinary skill in the art at the time of invention to modify the system of Misra et al. with Fulvio by using the features, as taught by Eo et al., in order to provide the logic unit includes: adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits; and combining means for combining the binary output values into the code bit, the combining means includes means for performing binary XOR operations, includes the steps of: performing binary AND operations, wherein each operation is adapted to combine a bit of the modified index (j) and a bit of the counter value (i), and to output a binary output value representing a binary AND combination of the two bits; and combining the binary output values into the code bit and regarding. The motivation of using these functions is to enhance the system in a cost effective manner.

8. Claims 5-6 and 15-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Misra et al. (US 2002/0009127 A1) in view of Fulvio (WO 01/50658 A1) as applied to claims 1-12 above, and further in view of Kim et al. (USP 6,671,251 B1).

Misra et al. and Fulvio disclose the claimed limitations as described in paragraph 5 above. Misra et al. and Fulvio do not disclose the following features: regarding claim 17, wherein step a) includes the steps of: permuting the bits of the index (k); and selecting, depending upon a mode signal indicating a desired type of the orthogonal code, the permuted index or the shifted index, thereby generating the modified index (j).

Kim et al. discloses a communication system for generating complex quasi-orthogonal code for spreading channel data with the following features: regarding claim 17, wherein step a) includes the steps of: permuting the bits of the index (k) (Fig. 5, a procedure for generating complex quasi-orthogonal codes, see "the M-sequence are column permuted with permutation function to generate a Walsh code" recited in column 5 lines 30-35 and column 8 lines 36-45), selecting, depending upon a mode signal indicating a desired type of the orthogonal code (Fig. 10, complex quasi-orthogonal code generator which generates quasi-orthogonal code mask in number, see "selectively outputs a mask corresponding to quasi-orthogonal code" recited in column 22 lines 12-20) and the permuted index or the shifted index, thereby generating the modified index (j) (Fig. 5, a procedure for generating complex quasi-orthogonal codes, see "by circularly shifting M-sequence and column permutating the circularly shifted specific sequences for converting the generated M-sequence to Walsh orthogonal codes" recited in column 2 lines 53-65).

It would have been obvious to one of the ordinary skill in the art at the time of invention to modify the system of Misra et al. with Fulvio by using the features, as taught by Kim et al., in order to provide the index conversion unit includes: a permutation unit for permuting the bits of the index (k); and selection means for selecting, depending upon a mode signal indicating a desired type of the orthogonal code, the output of the permutation unit or the output of the shift register, thereby generating the modified index (j), permuting the bits of the index (k) and permuting the bits of the index (k); and selecting, depending upon a mode signal indicating a desired type of the orthogonal code, the permuted index or the shifted index, thereby generating the modified index (j). The motivation of using these functions is to enhance the system in a cost effective manner.

9. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Misra et al. (US 2002/0009127 A1) in view of Fulvio (WO 01/50658 A1), Piccinonno (WO 01/50659 A1), Jechoux et al. (US 2002/0041636 A1), and further in view of Eo et al. (US 6,069,574) and further in view of Schooler et al. (US 2005/0053049 A1).

Misra et al. discloses the following features: regarding claim 21, generate an orthogonal code having a spreading factor (SF) and an index (k), (Fig. 5, a system structure, see “the spreader 17 comprises a code generator 21” recited in paragraph 0052 lines 1-7 and paragraph 0053 lines 1-7), wherein the spreading factor (SF) is selectable from values in a range 1<SF< or =.SFmax, with SFmax denoting a fixed maximum spreading factor wherein the index (k) is in a range 0<k< or =SFmax-1, wherein the computer program product performs the following steps when run on a processor converting by an index conversion unit the index (k) into a modified index (j) associated with a corresponding code having the fixed maximum spreading wherein the modified index (i) is in a range 0<i<or =SFmax-1 (Fig. 6a-d, control flow diagrams, see “form a periodic long code to the maximum spreading factor (SFmax) of the communication system” recited in paragraph 0064 lines 1-13 and paragraph 0072 line 1).

Misra et al. does not disclose the following features: regarding claim 21, a computer-readable medium encoded with a program product for, the converting step including multiplication means for multiplying the index (k) with a value of SF.sub.max/SF, by mapping unit for mapping the spreading factor (SF) to a number (s) equal to log.sub.2{SF.max/SF}, wherein the logic operations include performing binary AND operations, wherein the logic unit receive a bit of the modified index (j) and a bit of the counter value (i), and outputs a binary output value representing a binary AND combination of the two bits and combining the binary output values into the code bit, performing a shift register operation comprising receiving and storing the index (k) in binary representation and receiving the number (s) and to shift the stored index (k) by (s) bit positions in the direction of more significant bit positions, initializing a counter value (i), solely performing logic operations on bits of the modified index (j) and bits of the counter value (i), thereby generating a code bit of the orthogonal code, incrementing the counter value (i) by one and repeating steps c) and d) until a desired number of code bits has been generated.

Fulvio discloses the following features: regarding claim 21, initializing a counter value (i) solely performing logic operations on bits of the modified index (j) and bits of the counter value (i), thereby generating a code bit of the orthogonal code (Fig. 1, generator for orthogonal spreading sequence, see “the logical unit AND has two input valves of index k and counter value" recited in lines 19-24 page 1), incrementing the counter value (i) by one(Fig. 1, generator for orthogonal spreading sequence, see “the desired sequence is generated when counter runs" recited in lines 24-25 page 1) and repeating steps until a desired number of code bits has been generated (Fig. 1, generator for orthogonal spreading sequence, see “the desired sequence is generated when counter runs" recited in lines 24-25 page 1)

Misra et al. and Fulvio do not disclose the following features: regarding claim 21, a computer-readable medium encoded with a program product for, the converting step including multiplication means for multiplying the index (k) with a value of SF.sub.max/SF, by mapping unit for mapping the spreading factor (SF) to a number (s) equal to log.sub.2{SF.max/SF}, wherein the logic operations include performing binary AND operations, wherein the logic unit receive a bit of the modified index (j) and a bit of the counter value (i), and outputs a binary output value representing a binary AND combination of the two bits and combining the binary output values into the code bit, performing a shift register operation comprising receiving and storing the index (k) in binary representation and receiving the number (s) and to shift the stored index (k) by (s) bit positions in the direction of more significant bit positions.

Jechoux et al. disclose the following features; regarding claim 21, performing a shift register operation comprising receiving and storing the index (k) in binary representation (Fig. 3, bock diagram discrete in time of the transmission chain between the transmitter and the receiver, see "shift register having I boxes in series referenced by index k" recited in paragraph 0003 lines 6-9) and receiving the number (s) and to shift the stored index (k) by (s) bit positions in the direction of more significant bit positions (Fig. 3, bock diagram discrete in time of the transmission chain between the transmitter and the receiver, see "contents slide towards right each time a symbol arrives at the entry" recited in paragraph 0003 lines 9-11).

Misra et al., Fulvio and Jechoux et al. do not disclose the following features: regarding claim 21, a computer-readable medium encoded with a program product for, the converting step including multiplication means for multiplying the index (k) with a value of SF.sub.max/SF, by mapping unit for mapping the spreading factor (SF) to a number (s) equal to log.sub.2{SF.max/SF}, wherein the logic operations include performing binary AND operations, wherein the logic unit receive a bit of the modified index (j) and a bit of the counter value (i), and outputs a binary output value representing a binary AND combination of the two bits and combining the binary output values into the code bit.

Piccinonno discloses the following features: regarding claim 21, the converting step including multiplication means for multiplying the index (k) with a value of SF.sub.max/SF (Fig. 2, flow chart describing the a spreading sequence generator, see “a generator for rows of Hadamard matrices of dimension SF by an index n of k” recited in lines 21-27 on page 1) and by mapping unit for mapping the spreading factor (SF) to a number (s) equal to log.sub.2{SF.max/SF} (Fig. 2, flow chart describing the a spreading sequence generator, see “if necessary n is transformed as described obtaining the binary number” recited in lines 10-14 on page 3).

Misra et al., Fulvio, Jechoux et al. and Piccinonno do not disclose the following features: regarding claim 21, a computer-readable medium encoded with a program product for, wherein the logic operations include performing binary AND operations, wherein the logic unit receive a bit of the modified index (j) and a bit of the counter value (i), and outputs a binary output value representing a binary AND combination of the two bits.

Eo et al. disclose the following features: regarding claim 21, wherein the logic operations include performing binary AND operations (Fig. 1, a circuit for generating Hadamard codes, see “final Hadamard code is generated through a binary addition operation at AND gate” recited in column 1 lines 58-63), wherein the logic unit receive a bit of the modified index (j) and a bit of the counter value (i), and outputs a binary output value representing a binary AND combination of the two bits (Fig. 1, a circuit for generating Hadamard codes, see “two AND gates 118 and 119 receive output signal from the counter 112 and index bit from the register 114” recited in column 1 lines 41-44) and combining the binary output values into the code bit (Fig. 1, a circuit for generating Hadamard codes, see “a XOR gate 124hereby generating a Hadamard code” recited in column 1 lines 44-46).

Misra et al., Fulvio, Jechoux et al., Piccinonno and Eo et al. do not disclose the following features: regarding claim 21, a computer-readable medium encoded with a program product for.

Schooler et al. discloses a wireless system for a programmable code generator with the following features: regarding claim 21, a computer-readable medium encoded with a program product for (Fig. 3, generation of the synchronization channel (SCH), the primary common control physical channel (PCCPCH) and the common pilot channel (CPICH) by a node, see "computer usable medium including a program for generating the code sequence, see, “for selecting a code type, for generating a plurality of instruction and for storing the instruction” recited in paragraph 0021 lines 1-7).

It would have been obvious to one of the ordinary skill in the art at the time of invention to modify the system of Misra et al. with Fulvio by using the features, as taught by Schooler et al. et al. in order to provide a computer-readable medium encoded with a program product for. The motivation of using these functions is to enhance the system in a cost effective manner.

Response to Arguments

10. Applicant's arguments with respect to claims 1-2, 8, 10, 12-13, 17, 19 and 21 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

11. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SYED BOKHARI whose telephone number is (571)270-3115. The examiner can normally be reached on Monday through Friday 8:00-17:00 Hrs..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kwang B. Yao can be reached on (571) 272-3182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Syed Bokhari/
Examiner, Art Unit 2416
2/10/2009

/Kwang B. Yao/

Supervisory Patent Examiner, Art Unit 2416